

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Tinjauan Pustaka diperlukan untuk mengidentifikasi penelitian-penelitian serupa yang telah dilakukan sebelumnya, sehingga dapat melakukan perbedaan antara penelitian saat ini dengan penelitian-penelitian terdahulu. Tinjauan pustaka yang digunakan dalam penelitian ini merupakan penelitian-penelitian yang mengkaji tentang perancangan sistem penjualan dan persediaan barang. Berikut ini adalah beberapa penelitian yang digunakan sebagai tinjauan pustaka.

Pertama adalah Jurnal yang berjudul “Sistem Informasi Penjualan Dan Persediaan Barang Dagang Pada Perusahaan Hakasima Kota Ternate” (Burhan, 2018). Penelitian pada perusahaan Hakasima tersebut bertujuan untuk merancang sistem informasi penjualan dan persediaan barang dagang pada Perusahaan Hakasima Kota Ternate. Data transaksi dan persediaan pada perusahaan Hakasima masih menggunakan media yang manual dimana data persediaan dan transaksi dicatat secara satu persatu pada dokumen dan dijumlahkan pada kalkulator .

Metode pengumpulan data yang digunakan adalah observasi, wawancara dan dokumentasi kepada pihak perusahaan, Rancangan analisis sistem yang lama dan analisis sistem diusulkan menggunakan model *Flowchart* diagram serta hasil rancangan sistem menggunakan model *UML*, Bahasa pemrograman yang digunakan menggunakan *Borland Delphi 7* serta *Microsoft Acces* sebagai *database*, hasil peniltian ini dapat mempercepat pengelolaan data barang serta mempermudah dalam pembuatan catatan transaksi penjualan Barang serta pembuatan laporan pada sistem secara cepat.

Penelitian yang kedua dilakukan oleh Rahadiyan, dkk. (2018) yang berjudul Analisis Dan Perancangan Sistem Informasi Penjualan dan Persediaan Barang Pada Gudang pada CV. KAJEYEFood yang ditulis dalam jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer. Pada penelitian ini menganalisis dan merancang sistem informasi penjualan dan persediaan barang pada CV. KAJEYEFood dimana kegiatan pencatatan persediaan barang dilakukan secara

manual pada buku. Perancangan dilakukan dengan menggunakan *framework Rational Unified Process*. Analisis proses bisnis yang dilakukan dalam penelitian ini mencakup proses bisnis utama yang sedang berjalan pada CV. KAJEYEFood. Proses bisnis tersebut antara lain proses bisnis transaksi penjualan pada toko, pencatatan data barang pada gudang, dan pembuatan laporan. Pemodelan proses bisnis dilakukan dengan menggunakan BPMN. Proses bisnis usulan, dibuat dengan menggunakan sistem informasi yang dapat mendukung dan memudahkan pekerjaan pada bagian penjualan toko dan pencatatan data barang pada gudang. Pada bagian penjualan, proses perhitungan biaya transaksi pelanggan dan pembuatan nota penjualan kini dilakukan sistem yang mana dapat meningkatkan efisiensi waktu dan meningkatkan pelayanan pada pelanggan. Pada gudang, pencatatan data barang dilakukan dengan meng-inputkan data barang pada sistem, sehingga memudahkan proses pencarian data. Dan yang terakhir, proses pembuatan laporan dilakukan oleh sistem, dimana data laporan dapat dibuat berdasarkan jangka waktu tertentu.

Analisis persyaratan pada penelitian ini meliputi *vision document*, fitur, spesifikasi persyaratan fungsional dan *non fungsional*, pemodelan *use case*, *use case scenario* dan *activity diagram*. Hasil analisis yang didapatkan yakni berupa 8 fitur, 21 kebutuhan fungsional, 2 kebutuhan non fungsional, dan 9 *use case*. 3. Perancangan pada penelitian ini meliputi perancangan kelas analisis, pemetaan analisis, unifikasi kelas analisis, *class diagram*, data model, dan *sequence diagram*.

Evaluasi pada penelitian ini dilakukan dengan menggunakan dua metode yakni hasil evaluasi dengan *traceability matrix* menunjukkan bahwa setiap persyaratan sudah memiliki kode unik yang dapat dilacak hingga ke model diagram dan telah didefinisikan dengan benar dan hasil evaluasi menggunakan *consistency analysis* ini adalah RCI (*Requirement Consistency Index*) yang mempunyai nilai sebesar 100%. Hal ini menunjukkan bahwa pendefinisian kebutuhan yang telah dilakukan sudah 100% konsisten.

Penelitian yang ketiga dilakukan oleh Sikumbang, 2016), yang berjudul Sistem Informasi Persediaan Barang Berbasis *Website* Menggunakan Metode

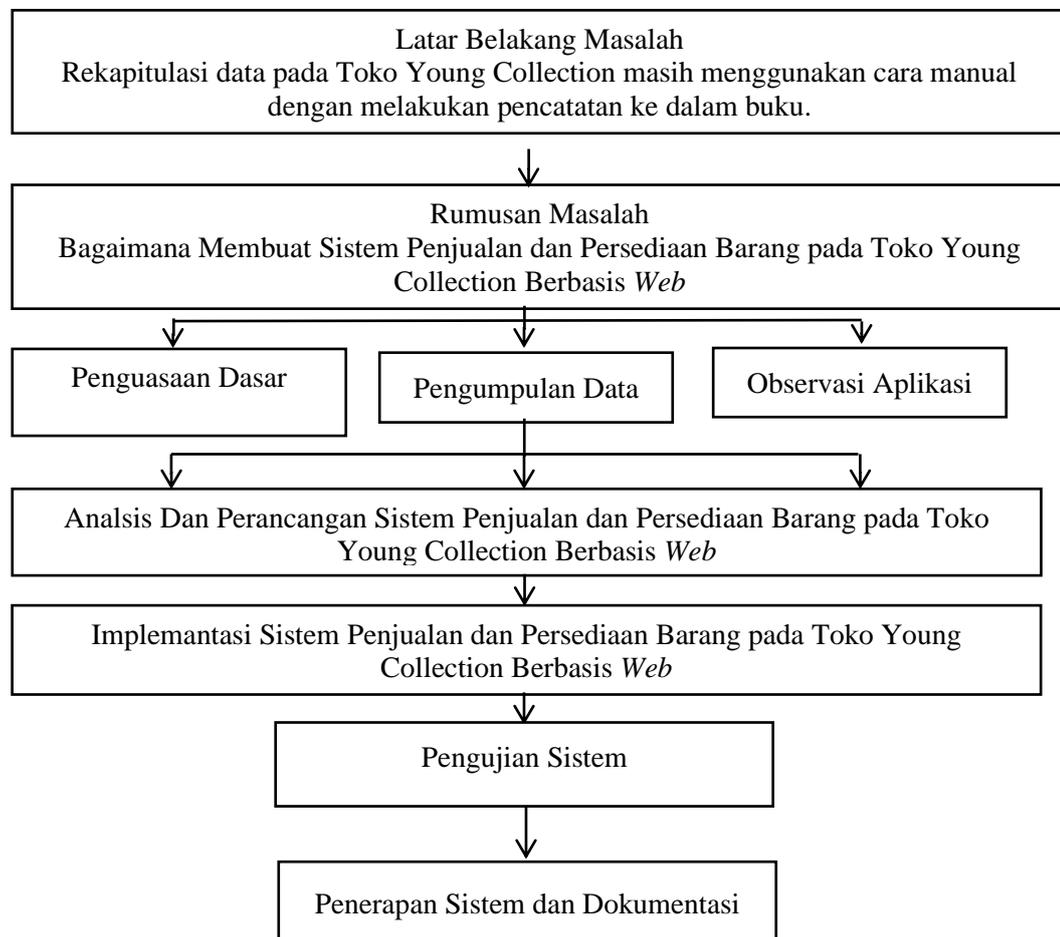
Waterfall Dengan Konsep Pemrograman Terstruktur dalam jurnal Seminar Nasional Ilmu Pengetahuan dan Teknologi Komputer Nusa Mandiri.

Sistem persediaan barang yang dilakukan pada saat ini masih menggunakan cara manual, dimana pencatatan data masih menggunakan kertas sehingga kurang terjamin keakuratan data, kemungkinan terjadi kesalahan pencatatan dan perhitungan atas transaksi yang terjadi, kesulitan dalam mencari data yang dibutuhkan dan mengontrol stok barang. Perancangan sistem informasi persediaan barang berbasis web ini menggunakan bahasa pemrograman *PHP* didukung dengan database *MySQL*. Model pengembangan sistem yang digunakan adalah model *waterfall*, analisis dan desain menggunakan diagram yang terkandung di dalam *UML*. Dengan sistem informasi persediaan barang berbasis *web* ini dapat mengurangi resiko kesalahan informasi dalam pencatatan persediaan barang, mempercepat pembuatan laporan dan membantu dalam menghasilkan keputusan-keputusan yang akurat dan cepat sehingga pelayanan terhadap pelanggan dapat meningkat dan membaik.

Berdasarkan beberapa penelitian yang telah dilakukan tersebut, dapat disimpulkan bahwa permasalahan yang sering muncul dalam proses bisnis yang terjadi dalam sebuah toko adalah pada bagian penjualan barang, pembuatan laporan, dan permasalahan pada bagian persediaan barang. Oleh karena itu dalam penelitian, dibangun sistem informasi untuk Toko Young Collection yang dapat menangani permasalahan tersebut. Dalam pembangunan sistem informasi ini juga ditambahkan fungsi untuk menangani masalah penjualan dan persediaan barang laporan penjualan, laporan persediaan barang ada dalam sistem penjualan dan persediaan barang pada Toko Young Collection.

2.2 Kerangka Pemikiran

Berikut ini adalah tahapan kerangka pemikiran dalam melakukan perancangan Sistem Penjualan dan Persediaan Barang pada Toko Young Collection Berbasis *Web*. Kerangka pemikiran dapat dilihat pada Gambar 2.1.



Gambar 2.1. Kerangka Pemikiran

Penjelasan dari kerangka pemikiran tersebut adalah :

a) Latar Belakang Masalah

Saat proses transaksi penjualan dan persediaan barang pada Young Collection semua data dicatat dalam buku. Dengan metode pencatatan dalam buku ini sering terjadi kesalahan dalam pengolahan data transaksi penjualan seperti sering terjadinya kesalahan pencatatan maupun jumlah perhitungan jumlah penjualan, lamanya penelusuran atau pencarian data penjualan, dan lamanya proses rekapitulasi data penjualan saat pembuatan laporan penjualan. Selain masalah dalam hal penjualan permasalahan muncul dalam data persediaan barang seperti seringnya data jumlah barang dalam buku tidak sesuai dengan jumlah

barang yang sebenarnya, sulit dan lamanya pencarian lamanya proses rekapitulasi data barang pada saat laporan persediaan barang.

b) Rumusan Masalah

Berdasarkan latar belakang masalah di atas, maka dapat dirumuskan masalah yaitu “Bagaimana membuat sistem Penjualan dan Persediaan barang pada Toko Young Collection berbasis *web*?”

c) Penguasaan dasar *PHP* dan *MySQL*

Penguasaan dasar *PHP* dan *MySQL* merupakan suatu kunci terciptanya sistem Penjualan dan Persediaan Barang pada toko Young Collection Berbasis *Web* Tersebut, karena pembuatan sistem Penjualan dan Persediaan Barang pada Toko Young Collection menggunakan bahasa pemrograman *PHP* dan *software MySQL* sebagai *database*.

d) Pengumpulan data

Tahap pengumpulan data pada penelitian ini melalui observasi langsung di Toko Young Collection serta wawancara dengan karyawan Young Collection mengenai permasalahan yang dibahas yaitu tentang proses penjualan dan persediaan barang. Pengumpulan data bertujuan untuk memperoleh dan mengumpulkan data yang dibutuhkan.

e) Observasi aplikasi

Observasi aplikasi bertujuan untuk mencari gambaran sistem yang akan dibuat dengan melihat beberapa aplikasi yang sudah ada.

f) Analisis dan perancangan sistem Penjualan dan Persediaan Barang pada Toko Young Collection berbasis *Web*.

Analisis dan perancangan sistem Penjualan dan Persediaan Barang pada Toko Young Collection berbasis *Web* bertujuan membuat perancangan sistem baru yang diharapkan dapat membantu menyelesaikan suatu pekerjaan agar lebih efektif dan efisien. Perancangan sistem pada penelitian ini menggunakan UML (*Unified Modeling Language*).

g) Implementasi sistem Penjualan dan Persediaan Barang pada Toko Young Collection berbasis *Web*.

Implementasi sistem Penjualan dan Persediaan Barang pada Toko Young Collection berbasis *Web* menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai *database*-nya.

h) Pengujian Sistem

Pengujian sistem merupakan tahap setelah sistem berhasil dibuat. Pengujian sistem bertujuan untuk mengetahui kelebihan, kelemahan serta mengetahui kelayakan suatu sistem untuk digunakan. Pengujian sistem pada penelitian ini menggunakan metode *WebQual*.

i) Penerapan Sistem dan Dokumentasi

Tahap penerapan sistem disini berarti sistem Penjualan dan Persediaan barang telah lolos uji aplikasi dan telah digunakan pada Young Collection untuk membantu menyelesaikan suatu pekerjaan, serta membuat dokumentasi dari seluruh kegiatan penyusunan tugas akhir.

2.3 Teori Pendukung

2.3.1 Sistem

1. Definisi Sistem

Menurut (Romney & Steinbart, 2015) mengungkapkan bahwa Sistem adalah suatu rangkaian yang terdiri dari dua atau lebih komponen yang saling berhubungan dan saling berinteraksi satu sama lain untuk mencapai tujuan dimana sistem biasanya terbagi dalam sub sistem yang lebih kecil yang mendukung sistem yang lebih besar.

2. Karakteristik Sistem

Menurut Hutahaen (2014) Supaya Sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu :

a. Komponen

Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan.

Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

b. Batasan Sistem (*boundary*)

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai suatu kesatuan. Batasan suatu sistem menunjukkan ruang lingkup (*scope*) dari sistem tersebut.

c. Lingkungan Luar Sistem (*environment*)

Lingkungan Luar Sistem adalah diluar batasan sistem yang mempengaruhi operasi sistem. Lingkungan dapat bersifat menguntungkan yang harus tetap di jaga dan yang merugikan yang harus dijaga dan dikendalikan, kalau tidak akan mengganggu kelangsungan hidup dari sistem.

d. Penghubung sistem (*Interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

e. Masukan Sistem (*Input*)

Masukan adalah energi yang dimasukkan kedalam sistem, yang dapat berupa perawatan (*Maintenance Input*), dan masukan sinyal (*Signal Input*). *Maintance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran. Contoh dalam sistem *computer program* adalah *maintenance input* sedangkan data adalah *signal input* untuk diolah menjadi informasi.

f. Keluaran sistem (*Output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna sisa pembuangan. Contoh komputer menghasilkan panas yang merupakan sisa pembuangan, sedangkan informasi adalah keluaran yang dibutuhkan.

g. Pengolah Sistem (*Process*)

Suatu sistem menjadi bagian pengolah yang akan merubah masukan menjadi keluaran. Sistem produksi akan mengolah bahan baku menjadi bahan jadi, system akuntansi akan mengolah data menjadi laporan-laporan keuangan.

i. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Sasaran dari sistem sangat menentukan input yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

2.3.2 Analisis Sistem

Menurut Mulyani (2016) mengungkapkan bahwa analisis sistem merupakan suatu teknik penelitian terhadap sebuah sistem dengan menguraikan komponen-komponen pada sistem tersebut dengan tujuan untuk mempelajari komponen itu sendiri serta keterkaitannya dengan komponen lain yang membentuk sistem sehingga didapat sebuah keputusan atau kesimpulan mengenai sistem tersebut baik itu kelemahan ataupun kelebihan sistem.

2.3.3 Prinsip-prinsip Analisis sistem

Menurut Tyoso (2016) Prinsip-prinsip Analisis sistem adalah :

1. Mendefinisikan masalah. Masalah yang akan dipecahkan dengan sistem diatur berkenaan dengan lingkungan tempat sistem berinteraksi.
2. Menyatakan sasaran sistem. Tujuan umum dan khusus yang ingin dicapai yang berkaitan dengan keefektifan ditetapkan dan diumumkan kepada pihak-pihak yang berkepentingan.
3. Menetapkan batasan sistem (*system Boundaries*). Pembatas antara sistem yang baru dengan lingkungannya harus diperinci. Hubungan sistem (*interface*) yang berkaitan dengan masukan dan keluaran harus ditegaskan.
4. Menetapkan kendala sistem. Kendala pada sistem dan proses pengembangannya, seperti biaya dan jangka waktu untuk pengembangan sistem, harus dipastikan.
5. Dekomposisi Sistem. Sistem dipecah kedalam sub-sistem yang saling terkait dan berhubungan dengan lingkungannya. Hubungan antar subsistem ditentukan sehingga seorang analis sistem mampu melihat sistem terinci. Subsistem yang

berada pada tingkat bawah lah yang nantinya dirancang dan menjadi bagian sistem yang ditetapkan.

2.3.4 Perancangan Sistem

Menurut Hidayat, dkk., (2016) Perancangan adalah proses merencanakan segala sesuatu terlebih dahulu. Perancangan merupakan wujud visual yang dihasilkan dari bentuk-bentuk kreatif yang telah direncanakan. Langkah awal dalam perancangan desain bermula dari hal-hal yang tidak teratur berupa gagasan atau ide-ide kemudian melalui proses penggarapan dan pengelolaan akan menghasilkan hal-hal yang teratur, sehingga hal-hal yang sudah teratur bisa memenuhi fungsi dan kegunaan secara baik. Perancangan merupakan penggambaran, perencanaan, pembuatan sketsa dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi.

2.3.5 Pengertian Persediaan

Persediaan (*inventory*) merupakan barang yang dijual dalam aktivitas operasi normal perusahaan. Dengan pengecualian organisasi jasa tertentu, persediaan merupakan aset inti dan penting dalam perusahaan. Persediaan harus diperhatikan karena merupakan komponen utama dari aset operasi dan langsung mempengaruhi perhitungan laba (Kurniawan & Putri, 2016).

2.3.6 Pengertian Penjualan

Penjualan adalah proses akhir dari kegiatan pemasaran, karena pada proses ini ada penetapan harga, serah terima barang dan adanya pembayaran yang disepakati oleh penjual dan pembeli (Hayuningtyas , 2015).

2.3.7 Hypertext Markup Language HTML

HTML atau *Hypertext Markup Language* merupakan bahasa pemrograman yang bisa digunakan untuk membuat aplikasiberbasis *web*. Bahasa pemrograman ini ditulis dalam berkas format ASCII, supaya dapat menghasilkan tampilan wujud yang terintegrasi. HTML adalah bahasa pemrograman *web* yang umum dan bisa digunakan serta mudah dipelajari (Maya, dkk. 2015).

2.3.8 Website

Website adalah kumpulan halaman yang menampilkan informasi data teks, data gambar diam atau gerak, data animasi, suara, video atau gabungan dari semuanya yang baik untuk membentuk satu rangkaian bangunan yang saling terkait di mana masing-masing dihubungkan dengan jaringan-jaringan halaman (Faturrahman, 2014).

2.3.9 *Cascading Style Sheet* (CSS)

CSS (*cascading style sheet*) merupakan salah satu Bahasa pemrograman *web* yang digunakan untuk mempercantik halaman *web* dan mengendalikan beberapa komponen dalam sebuah *web* sehingga akan lebih terstruktur dan seragam (Bekti, 2015)

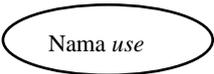
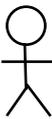
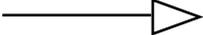
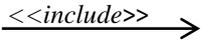
2.3.10 *Unified Modeling Language* (UML)

Menurut (Rosa & Shalahuddin, 2016) mengungkapkan bahwa pada perkembangan teknologi perangkat lunak diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (UML). UML adalah bahasa visual yang digunakan dalam pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML memiliki 13 diagram, seperti *class diagram*, *object diagram*, *component diagram*, *composite diagram*, *package diagram*, *deployment diagram*, *use case diagram*, *activity diagram*, *state machine diagram*, *sequence diagram*, *communication diagram*, *timing diagram*, dan *interaction diagram*.

2.3.11 Tipe-Tipe Diagram UML

a. Use case Diagram

Tabel 2.1 Simbol-simbol Use Case Diagram

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Use case</i>	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau <i>actor</i> .
2		<i>Actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
3		<i>Association</i>	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu.
4		<i>Generalization</i>	Hubungan generalisasi dan spesialisasi (umum – khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
5		<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

Use case Diagram atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui

fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol *use case diagram* dapat dilihat pada Tabel 2.1.

b. *Class Diagram*

Tabel 2.2. Simbol-simbol *Class Diagram*

NO	SIMBOL	NAMA	KETERANGAN			
1	<table border="1"> <tr> <td>Nama kelas</td> </tr> <tr> <td>+ Atribut</td> </tr> <tr> <td>+ Operasi()</td> </tr> </table>	Nama kelas	+ Atribut	+ Operasi()	<i>Class</i>	Kelas pada struktur sistem
Nama kelas						
+ Atribut						
+ Operasi()						
2		<i>Interface</i>	Sama dengan konsep interface dalam pemrograman berorientasi objek.			
3		<i>Association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .			
4		<i>Directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain biasanya juga disertai dengan <i>multiplicity</i> .			
5		<i>Generalization</i>	Relasi antar kelas dengan makna generalisasi - spesialisasi (umum – khusus).			
6		<i>Dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.			
7		<i>Aggregation</i>	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).			

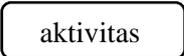
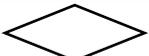
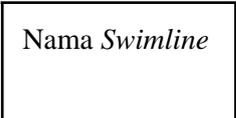
Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode operasi. Kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem sehingga pembuat perangkat lunak dapat membuat kelas-kelas di dalam program perangkat lunak

sesuai dengan perancangan diagram kelas, (Rosa & Shalahuddin, 2016). Simbol *class diagram* dapat dilihat pada Tabel 2.2.

c. Activity Diagram

Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa & Shalahuddin, 2016). Simbol *Activity Diagram* ditunjukkan pada Tabel 2.3.

Tabel. 2.3 Simbol-simbol *Activity Diagram*

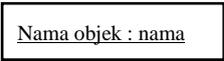
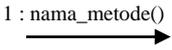
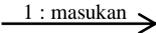
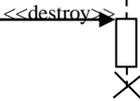
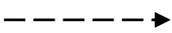
NO	GAMBAR	NAMA	KETERANGAN
1		Status awal	Status awal aktivitas sistem.
2		Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3		<i>Decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4		<i>Join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5		Status akhir	Status akhir yang dilakukan sistem.
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

d. Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga

dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang dibuat juga semakin banyak. (Rosa & Shalahuddin, 2016). Simbol *Sequence diagram* dapat dilihat pada Tabel 2.4.

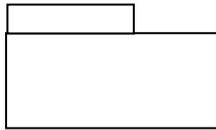
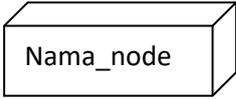
Tabel 2.4. Simbol-Simbol *Sequence Diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		Aktor	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri.
2		<i>Lifeline</i>	Menyatakan kehidupan suatu objek.
3		Objek	Menyatakan objek yang berinteraksi pesan.
4		Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi.
5		Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
6		Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri.
7		Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
8		Pesan tipe <i>destroy</i>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .
9		Pesan Tipe Return	Menyatakan bahwa suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i>

e. *Deployment Diagram*

Diagram *deployment* atau *deployment diagram* menunjukkan konfigurasi komponen dalam proses eksekusi aplikasi. Simbol *deployment diagram* dapat dilihat pada Tabel 2.5.

Tabel 2.5. Simbol-Simbol *Deployment diagram*

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Package</i>	Package merupakan suatu bungkus dari satu atau lebih node
2		<i>Node</i>	Biasanya mengacu pada <i>hardware</i> , perangkat lunak yang tidak dibuat sendiri, jika didalam <i>node</i> disertakan komponen untuk mengkonsistensikan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
3		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
4		<i>Link</i>	Relasi antar <i>node</i>

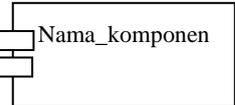
f. *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. Diagram komponen fokus pada komponen sistem yang dibutuhkan dan ada didalam sistem. Komponen dasar yang biasanya ada dalam suatu sistem adalah sebagai berikut:

- Komponen *user interface* yang menangani tampilan
- Komponen *bussiness processing* yang menangani fungsi-fungsi proses bisnis
- Komponen data yang menangani manipulasi data
- Komponen *security* yang menangani keamanan sistem

Simbol *component diagram* dapat dilihat pada Tabel 2.6.

Tabel 2.6. Simbol-Simbol Component diagram

NO	SIMBOL	NAMA	KETERANGAN
1		<i>Package</i>	Package merupakan suatu bungkusannya dari satu atau lebih node
2		Komponen	Komponen sistem
3		<i>Dependency</i>	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
4		<i>Link</i>	Relasi antar <i>node</i>
5		<i>Interface</i>	Sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen agar tidak mengakses langsung komponen

2.3.12 Database

Database adalah kumpulan tabel walaupun lebih tepat dikatakan kumpulan objek karena yang terkandung di dalam *database* sebenarnya bukan hanya tabel, melainkan indeks, *view*, *constraint* dan *trigger* (Raharjo, 2015).

Ada beberapa operasi *database* diantaranya :

1. Pembuatan basis data baru (*create database*), yang identik dengan pembuatan lemari arsip yang baru.
2. Penghapusan basis data (*drop database*), yang identik dengan perusakan lemari arsip (sekaligus beserta isinya jika ada).
3. Pembuatan file/tabel baru ke suatu basis data (*create table*), yang identik dengan penambahan map arsip baru ke sebuah lemari arsip yang telah ada.
4. Penghapusan file/tabel dari suatu basis data (*drop table*), yang identik dengan perusakan map arsip lama yang ada di sebuah lemari arsip.

5. Penambahan/pengisian data baru ke sebuah file/tabel di sebuah basis data (*insert*), yang identik dengan penambahan lembaran arsip ke sebuah map arsip.
6. Pengambilan data dari sebuah file/tabel (*retrieve/search*), yang identik dengan pencarian lembaran arsip dari sebuah map arsip.
7. Pengubahan data dari sebuah file/tabel (*update*), yang identik dengan perbaikan isi lembaran arsip yang ada di sebuah map arsip.
8. Penghapusan data dari sebuah file/tabel (*delete*), yang identik dengan penghapusan sebuah lembaran arsip yang ada di sebuah map arsip.

2.3.13 My Structure Query Language (MySQL)

MySQL (My Structure Query Language) adalah *software* RDBMS yang dapat mengolah *database* dengan cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user dan dapat melakukan suatu proses secara sinkron atau berbarengan (Raharjo, 2015).

2.3.14 Bootstrap

Bootstrap adalah paket aplikasi siap pakai untuk membuat *front-end* sebuah website. Bisa dikatakan, *bootstrap* adalah template desain *web* dengan fitur plus. *Bootstrap* diciptakan untuk mempermudah proses desain *web* bagi berbagai tingkat pengguna, mulai dari level pemula hingga yang sudah berpengalaman. Cukup bermodalkan pengetahuan dasar mengenai HTML dan CSS, anda pun siap menggunakan *bootstrap* (Rozi & Dev, 2015).

2.3.15 Hypertext Preprocessor (PHP)

PHP (*Hypertext Preprocessor*) adalah suatu bahasa pemrograman yang digunakan untuk menerjemahkan baris kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang berbasis server-side yang dapat ditambahkan ke dalam HTML (Supono & Putratama, 2016).

2.4 Pengujian WebQual

Menurut (Rosa & Shalahuddin, 2016) pengujian perangkat lunak adalah sebuah elemen topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi (*verification*) dan validasi (*validation*). Verifikasi mengacu pada sekumpulan aktifitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi

mengacu pada sekumpulan aktifitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan (*customer*). Penelitian ini menggunakan model pengujian *WebQual*.

Menurut Iman Sanjaya dalam jurnal Penelitian IPTEK-KOM, *WebQual* merupakan salah satu metode atau teknik pengukuran kualitas *website* berdasarkan persepsi pengguna akhir. Metode ini merupakan pengembangan dari SERVQUAL yang banyak digunakan sebelumnya pada pengukuran kualitas jasa. *WebQual* sudah mulai dikembangkan sejak tahun 1998 dan telah mengalami beberapa interaksi dalam penyusunan dimensi dan butir pertanyaannya.

Menurut teori *WebQual* pada jurnal yang disusun oleh (Anwariningsih), terdapat tiga dimensi yang mewakili kualitas suatu *website*, yaitu kegunaan (*usability*), kualitas informasi (*informationquality*) dan interaksi layanan (*serviceinteraction*). Masing-masing dimensi terdiri dari beberapa pernyataan yang ditunjukkan pada Tabel 2.5 sampai Tabel 2.8.

Tabel 2.7 Dimensi Kegunaan (*usability*)

No	Deskripsi Indikator
1	<i>Website</i> mudah untuk dioperasikan.
2	Interaksi dengan <i>website</i> sangat mudah dimengerti dan tidak membingungkan.
3	Pengguna merasa mudah untuk bernavigasi dalam <i>website</i> .
4	<i>Website</i> mudah untuk digunakan.
5	<i>Website</i> memiliki tampilan yang menarik.
6	Desain <i>website</i> sesuai dengan tipe <i>website</i> .
7	<i>Website</i> menunjukkan kemampuannya.
8	<i>Website</i> dapat memberikan pengaruh/ pengalaman positif bagi pengguna.

Tabel 2.7 Dimensi Kualitas Interaksi (*interaction quality*)

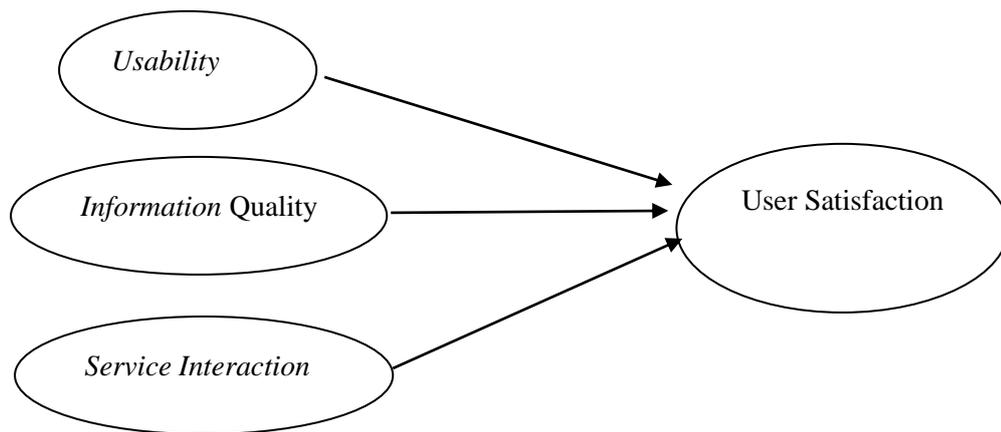
No	Deskripsi Instansi
1	<i>Website</i> memiliki reputasi yang baik.
2	Pengguna merasa aman untuk melakukan transaksi atau interaksi dengan <i>website</i> ..
3	<i>Website</i> menjaga informasi pribadi pengguna.
4	<i>Website</i> memberi ruang untuk personalisasi.
5	<i>Website</i> memberi ruang untuk komunitas.
6	<i>Website</i> memudahkan pengguna untuk berkomunikasi dengan organisasi.
7	Pengguna merasa yakin dengan layanan/informasi yang disediakan karena sesuai dengan yang dijanjikan.

Tabel 2.8 Dimensi Kualitas Informasi (*information quality*)

No	Deskripsi Indikator
1	<i>Website</i> menyajikan informasi yang akurat.
2	Informasi yang disajikan <i>website</i> dapat dipercaya.
3	Informasi yang disajikan tepat waktu atau <i>uptodate</i> .
4	Informasi yang disajikan relevan.
5	Informasi yang disajikan mudah dipahami.
6	Informasi yang disajikan sangat detail.
7	Informasi disajikan dalam format yang sesuai.

Tabel 2.9 Uji Keseluruhan

No	Deskripsi Indikator
1	Deskripsi Secara Umum Tentang <i>Website</i> Ini



Gambar 2.2 Hipotesis Penelitian (Silalahi, 2015)

Beberapa tahapan yang harus dilakukan untuk melakukan pengujian *webqual* antara lain:

2.4.1 Uji Instrument

Menurut Prakosad (2017) mengungkapkan bahwa uji instrument digunakan untuk mengetahui deskripsi mengenai variabel-variabel dalam penelitian, uji instrument terdiri dari uji validitas dan reliabilitas.

2.4.1.1 Uji Validitas

Menurut Arikunto (2010) mengungkapkan bahwa validitas adalah suatu ukuran yang menunjukkan tingkat kehandalanan kesahihah suatu instrument. Instrument yang valid berarti alat ukur yang digunakan untuk mendapatkan data itu valid. Menurut (Sugiyono, 2010) valid berarti instrument tersebut dapat digunakan untuk mengukur apa yang hendak diukur. Suatu kuisisioner dinyatakan valid apabila r hitung $>$ r tabel, sebaliknya kuisisioner dapat dinyatakan tidak valid

apabila $r_{hitung} < r_{tabel}$. Pengujian validitas ini menggunakan batasan r_{tabel} dengan signifikansi 0,05 dan uji 2 sisi.

2.4.1.2 Uji Reliabilitas

Menurut Arikunto (2010) mengungkapkan bahwa reliabilitas adalah suatu instrument yang dapat dipercaya untuk digunakan sebagai alat pengumpul data karena instrument tersebut sudah baik. Instrumen yang reliabel atau dapat dipercaya akan menghasilkan data yang dapat dipercaya, instrument yang reliabel mengandung arti bahwa instrument tersebut harus baik sehingga mampu mengungkap data yang bias dipercaya.

Menurut Ghazali (2012) mengungkapkan bahwa terdapat aturan praktis yang dapat diterapkan terkait dengan nilai alpha, jika $\alpha > 0,9$ berarti reliabilitas model sangat bagus, $\alpha > 0,8$ berarti reliabilitas model bagus, $\alpha > 0,7$ artinya reliabilitas model bias diterima, $\alpha > 0,6$ berarti model reliabilitas layak, $\alpha > 0,5$ berarti model reliabilitas kurang bagus, dan $\alpha < 0,5$ berarti reliabilitas model dapat diterima.

2.4.2 Uji Asumsi Klasik

Menurut (Yoedo, Susilowati & Julianto, 2016) uji asumsi klasik digunakan sebelum uji regresi linier berganda, ada data yang harus terpenuhi agar kesimpulan dari regresi bisa menjadi kuat, antara lain uji normalitas, uji multikolinearitas, uji heteroskedastisitas dan uji autokorelasi.

2.4.2.1 Uji Normalitas

Menurut Prakosad (2017) uji normalitas bertujuan untuk menguji apakah dalam model regresi, variabel pengganggu atau residual memiliki distribusi secara normal. Uji t dan f mengasumsikan bahwa residual mengikuti distribusi normal, kalau asumsi ini dilanggar maka uji statistic menjadi tidak valid.

Salah satu cara uji statistic one-sample kolmogorov-smirnov. Dasar pengambilan keputusan dari one-sample kolmogorov-smirnov adalah:

1. Jika hasil *one sample kolmogorov-smirnov* diatas tingkat signifikasi 0,05 menunjukkan pola distribusi normal, maka model regresi tersebut memenuhi asumsi normalitas.
2. Jika hasil *one-sample* dibawah *Kolmogorov-smirnov* dibawah tingkat signifikasi 0.05 tidak menunjukkan pola distribusi normal, maka model tersebut tidak memenuhi asumsi normalitas.

2.4.2.2 Uji Multikolinearitas

Menurut Prakosad (2017) pengujian multikolinearitas bertujuan untuk menguji apakah model regresi ditemukan adanya korelasi antara variabel bebas (*independent*). Model regresi yang baik seharusnya tidak ada korelasi diantara variabel *independent*. Jika variabel *independent* saling berkorelasi, maka variabel-variabel ini tidak orthogonal.

Variabel orthogonal merupakan variabel *independent* yang nilai korelasi antar sesama variabel *independent* sama dengan nol. Salah satu cara untuk mengetahui ada tidaknya multikolonearitas pada suatu model regresi adalah dengan melihat nilai *tolerance* dan VIF (*Variance Inflation Factor*).

1. Jika nilai *tolerance* > dan VIF < 10,0 maka dapat diartikan bahwa tidak terdapat multikolinearitas pada penelitian tersebut.
2. Jika nilai *tolerance* < 0.10 VIF > 10 maka dapat diartikan bahwa terjadi gangguan multikolinearitas pada penelitian tersebut.

2.4.2.3 Uji Heterokedastisitas

Menurut Prakosad (2017) pengujian ini bertujuan untuk menguji apakah dalam model regresi terjadi ketidaksamaan varians dari residual satu pengamatan ke pengamatan yang lain. Jika varians dari residual satu pengamatan ke pengamatan yang lain tetap. Maka disebut homoskedastisitas dan jika berbeda disebut heterokedastisitas. Untuk mengetahui adanya heterokedastisitas adalah dengan melihat ada/tidaknya pola tertentu pada grafik scatter plot dengan ketentuan:

1. jika terdapat pola tertentu, seperti titik-titik yang membentuk pola tertentu yang teratur maka menunjukkan telah terjadi heterokedastisitas.
2. Jika tidak ada pola yang kelas, serta titik – titik menyebar diatas dan bawah angka 0 pada sumbu Y maka tidak terjadi heterokedastisitas.

2.4.3 Analisis Regresi Linear Berganda

Menurut Prakosad (2017) mengungkapkan bahwa metode analisis regresi linear berganda berfungsi untuk mengetahui pengaruh/hubungan variabel *independent* (bebas) dengan variabel *dependent* (terikat). Pengujian regresi linear berganda ini dilakukan dengan melakukan uji koefisien determinasi, Uji Persial (Uji T) dan Uji Simultan (Uji F).

2.4.3.1 Uji F (Uji Simultan)

Menurut Ghozali (2011) mengungkapkan bahwa Uji F digunakan untuk menunjukkan apakah semua variabel independent yang dimasukkan dalam model mempunyai pengaruh secara Bersama-sama terhadap variabel *dependent*. Dasar penerimaan atau penolakan hipotesis dapat dilihat dengan membandingkan F hitung dengan F tabel, jika F hitung > F tabel maka Ho ditolak dan Ha diterima.

2.4.3.2 Uji T (Uji Persial)

Menurut Ghozali (2011) mengungkapkan bahwa uji T pada dasarnya menunjukkan seberapa jauh pengaruh satu variabel *independent* secara individual dalam menerangkan variabel *dependent*. Penerimaan atau penolakan hipotesis dilakukan dengan kriteria sebagai berikut:

1. Jika nilai $\text{sig} < 0.05$ atau T hitung > T tabel maka terdapat pengaruh X (variabel bebas) terhadap variabel Y (variabel terikat)
2. Jika nilai $\text{sig} > 0,05$ atau T hitung < T table maka tidak terdapat pengaruh variabel X (variabel bebas) terhadap variable Y (variabel terikat).

2.4.3.3 Koefisien Determinasi (R^2)

Menurut Prakosad (2017) mengungkapkan bahwa koefisien determinasi (R^2) pada intinya mengukur seberapa jauh kemampuan model menerangkan variasi variabel *dependent*. *Range* nilainya antara 0-1, apabila nilai R^2 kecil berarti kemampuan variabel-variabel independent dalam menjelaskan variasi variabel *dependent* sangat terbatas, dan sebaliknya apabila R^2 besar berarti kemampuan variabel-variabel *independent* dalam menjelaskan variasi variabel *dependent* besar.